Практическое задание №12
(классы, дружественные функции, перегрузка операторов)

Создать проект **Task12** с реализацией класса (классов) согласно своему варианту. Класс должен содержать **счётчик числа существующих объектов**, конструктор по умолчанию, конструктор (или конструкторы) с параметрами, конструктор копирования, конструктор переноса, деструктор, методы получения и изменения закрытых данных. Должны быть перегружены оператор присваивания с копированием, оператор присваивания с переносом, необходимые арифметические операторы, операторы сравнения, преобразования типа, индексирования, вызова функции, а также потоковые операторы вставки и извлечения.

Для сдачи проекта использовать структуру в файловой системе:

gxxxxx/12/Task12/ClassName.h – файл с объявлением класса.
gxxxxx/12/Task12/ClassName.cpp – файл с определением методов класса.
gxxxxx/12/Task12.Tests/ClassName.Tests.cpp – файлы с тестами пунктов задания.

Необходимые условия сдачи:

- стиль программирования должен соответствовать установленным правилам;
- программа должна быть протестирована с помощью gtest.

Вариант 1

Класс **Date** для работы с датами. Дата представляется тремя полями: год, месяц и день. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), операции: вычисление даты через заданное количество дней (с помощью перегруженного оператора +), вычитание заданного количества дней из даты (с помощью перегруженного оператора −), определение високосного года, присвоение и получение отдельных частей даты (год, месяц, день), сравнение дат (равно, до, после), вычисление количества дней между датами. Предусмотреть поле со статусом даты (“WINTER”, “SPRING”, “SUMMER”, “AUTUMN”) в виде C-строки (char *) и его изменение с изменением даты. (Указание: каждый год, число которого делится на 4, является високосным кроме годов, числа которых делятся на 100, но не делятся на 400. Например, 1700 и 1800 годы не являются високосными.)

Вариант 2

Класс **Address**, содержащий информацию о почтовом адресе организации. Поля данных должны включать в виде C-строк (char *) страну, город, улицу, номер дома и название организации. Предусмотреть возможность раздельного изменения составных частей адреса. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), перегруженные операторы сравнения объектов по названию организации, по частям адреса (страна, город). Создать реестр организаций в виде массива объектов класса Address и организовать поиск объектов в реестре по различным критериям (например, найти все организации заданной страны или все организации, расположенные в заданном городе).

Вариант 3

Класс **Cylinder** для представления цилиндра. Предусмотреть поле данных типа char * с названием фигуры в виде C-строки (строка должна содержать и номер объекта по порядку создания, например: “Цилиндр № 3” или “Cylinder № 3”), поля с координатами центра, радиусом основания, высотой цилиндра. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), виде перегруженных операторов: вычисление площади основания, вычисление длины описывающей

основание окружности, вычисление площади боковой поверхности цилиндра, вычисление объёма фигуры, а также сравнение разных цилиндров по объёму (в виде дружественной функции). Реализовать оператор сложения цилиндра с вещественным числом, подразумевая увеличение на эту величину высоты цилиндра (результатом оператора должен быть новый цилиндр, исходный «портить» нельзя).

Вариант 4

Класс **Point** для работы с точками на плоскости, заданными в полярных или декартовых координатах. Поля данных должны включать сами координаты и в виде С-строки (char *) тип представления точки (в полярных или декартовых координатах). Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), предусмотреть возможность изменения типа представления точки с преобразованием полярных координат в декартовы и наоборот. Должны быть реализованы: перемещение точки по оси X, перемещение по оси Y, определение расстояния между двумя точками (перегруженный оператор “–“ в виде дружественной функции), сравнение точек на совпадение и несовпадение.

Вариант 5

Класс **Abiturient**. Поля данных (типа char *) должны включать в виде С-строк фамилию, имя, отчество, в виде структуры домашний адрес (страна, город, улица, номер дома и номер квартиры), а также массив из 4 оценок по вступительным экзаменам. Предусмотреть возможность раздельного изменения ФИО и составных частей адреса. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), перегруженные операторы сравнения объектов по ФИО, по частям адреса (страна, город), метод вычисления средней оценки за вступительные экзамены. Создать список абитуриентов в виде массива объектов класса Abiturient и организовать поиск объектов в списке по различным критериям (например, найти всех абитуриентов из заданной страны или из заданного города, всех абитуриентов, средняя оценка которых не ниже заданной).

Вариант 6

Класс **Vector** (одномерный массив). Поля данных должны включать размер вектора и указатель на динамический одномерный массив вещественных элементов. Реализовать конструктор, создающий вектор требуемого размера с нулевыми значениями элементов, конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), явный конструктор преобразования (explicit), создающий вектор размера 10 с одинаковыми значениями элементов, равными переданному вещественному параметру, методы реверсирования и сортировки элементов вектора, а также в виде перегруженных операторов оператор сложения (+) векторов, оператор индексирования. Другие методы класса ввести по желанию.

Вариант 7

Класс **RusMoney** предназначен для работы с денежными суммами. Сумма должна быть представлена полями-номиналами, значениями которых должно быть количество купюр соответствующего достоинства. Номиналы российских рублей могут принимать значения 1, 2, 5, 10, 50, 100, 500, 1000, 5000. Копейки представить как 0.01 (1 копейка), 0.05 (5 копеек), 0.1 (10 копеек), 0.5 (50 копеек). Предусмотреть поле в виде С-строки (char *) со статусом суммы:

“MICRO”	0 – 999,99 руб.
“SMALL”	1000 – 9 999,99 руб.
“NORMAL”	10 000 – 99 999,99 руб.
“LARGE”	более 100 000 руб.

Статус суммы должен изменяться при её переходе из одного диапазона в другой. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с

копированием и с переносом), в виде перегруженных операторов сложение сумм, вычитание сумм, деление суммы на число, умножение на число, а также операции сравнения сумм и сравнения суммы с числом. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой.

Вариант 8

Класс **RingPayment**, представляющий собой разовый платеж за телефонный разговор. Класс должен содержать поля: номер телефона, тариф оплаты минуты разговора, скидка (в процентах), время начала разговора (в часах и минутах), продолжительность разговора (в минутах), сумма к оплате и категория разговора в виде С-строки (“DAY”, “PEAK” или “NIGHT”). Скидка формируется в зависимости от времени начала разговора (категории разговора). Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), метод вычисления суммы платежа за разговор, а также метод, определяющий «аналогичность» платежей (по одинаковому номеру телефона). В виде дружественной функции перегрузить оператор сложения (+) двух аналогичных (для одного номера телефона) платежей, возвращающий итоговую сумму, причитающуюся к оплате.

Вариант 9

Класс прямоугольников **Rectangle** со сторонами, параллельными осям координат. Поля данных должны включать название фигуры в виде С-строки (строка должна содержать и номер объекта по порядку создания, например: “Прямоугольник_4” или “Rectangle_4”), координаты точки левого верхнего угла, размеры прямоугольника по высоте и ширине. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), методы перемещения прямоугольников на плоскости, изменения размеров, построения наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Реализовать оператор сложения прямоугольника с числом, подразумевая увеличение на эту величину обоих его размеров (результатом оператора должен быть новый прямоугольник, исходный «портить» нельзя).

Вариант 10

Класс **Cone** для представления конуса Предусмотреть поле данных типа char * с названием фигуры в виде С-строки (строка должна содержать и номер объекта по порядку создания, например: “Конус#3” или “Cone#3”), поля с координатами центра, радиусом основания, высотой конуса. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), виде перегруженных операторов: вычисление площади основания, вычисление длины описывающей основание окружности, вычисление площади боковой поверхности конуса, вычисление объёма фигуры, а также сравнение разных конусов по объёму. Реализовать в виде дружественной функции оператор сложения конуса с вещественным числом, подразумевая увеличение на эту величину радиуса основания конуса (результатом оператора должен быть новый конус, исходный «портить» нельзя).

Вариант 11

Класс **BusRoute**. Поля данных должны включать номер маршрута, в виде С-строк (типа char *) пункт отправления, пункт назначения, время отправления, время прибытия на конечный пункт и цену проезда. Предусмотреть возможность раздельного изменения составных частей информации о маршруте. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), перегруженные операторы сравнения объектов по пункту отправления, пункту назначения, метод вычисления

продолжительности движения по маршруту. Создать класс **Schedule** (расписание автовокзала) в виде массива объектов класса BusRoute и организовать поиск объектов в массиве по различным критериям (например, найти все маршруты из заданного пункта отправления или в заданный пункт назначения, все маршруты, цена проезда по которым не превышает заданную и т.п.).

Вариант 12

Класс **ULongInteger** для представления беззнаковых длинных целых. Целое число должно быть представлено строкой символов в десятеричной системе счисления (поле класса типа char * – С-строка). Требуется реализовать: конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), арифметическую операцию сложения длинных целых посредством перегрузки оператора +, операцию сложения длинного целого с беззнаковым целым числом встроенного типа (например, unsigned int или size_t), перегруженные операторы сравнения длинных целых и длинного целого с беззнаковым целым числом встроенного типа.

Вариант 13

Класс **Vector3D**, задаваемый тремя координатами. Предусмотреть поле данных типа char * с названием вектора в виде С-строки (строка должна содержать и номер объекта по порядку создания, например: “Вектор-5” или “Vector-5”) и поля с координатами вектора. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), в виде перегруженных операторов: сложение и вычитание векторов, скалярное произведение векторов, умножение на скаляр, сравнение векторов, вычисление длины вектора, сравнение длин векторов (в виде дружественной функции).

Вариант 14

Класс **PCModel** – модель компьютера, который представляется полями: название марки компьютера, тип процессора, объём оперативной памяти, объём жёсткого диска, объём памяти видеокарты, стоимость компьютера и количество экземпляров, имеющихся в наличии. Для строковых данных предусмотреть поля типа char * (С-строка). Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), операции сравнения компьютеров по объёму оперативной памяти, объёму жёсткого диска, объёму памяти видеокарты, стоимости.

Создать класс **PCModelList** (список моделей компьютера) в виде массива произвольного числа объектов класса PCModel и организовать поиск по марке компьютера, типу процессора, объёму оперативной памяти, объёму жёсткого диска.

Вариант 15

Класс **Book** (книга). Поля данных должны включать год издания и в виде С-строк (типа char *) фамилию автора, название произведения. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом). Предусмотреть возможность раздельного изменения составных частей информации о книге. Создать класс **Library** (библиотека) в виде массива произвольного числа объектов класса Book и организовать поиск книги по какому-либо признаку (например, по автору или по году издания), добавление книги в библиотеку (с помощью перегруженного оператора +), удаление книги из неё (с помощью перегруженного оператора −), сортировки книг по разным полям.

Вариант 16

Класс **Time** для работы со временем. Время представляется тремя полями: часы, минуты и секунды. Предусмотреть поле типа char * в виде С-строки со статусом времени (“MORNING”, “DAY”, “EVENING”, “NIGHT”) и его изменением с изменением времени. Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), операции: вычисление разности между двумя моментами

времени (с помощью перегруженного оператора -), сложение времени и заданного количества секунд (с помощью перегруженного оператора +), вычитание из времени заданного количества секунд (с помощью перегруженного оператора −), сравнение моментов времени, перевод в секунды, перевод в минуты (с округлением до целой минуты).

Вариант 17

Класс **Modem** – модем, который представляется полями: название марки модема, название фирмы производителя, тактовая частота сигнального процессора, объём оперативной памяти, количество портов ввода-вывода, потребляемая мощность. Для строковых данных предусмотреть поля типа char * (С-строка). Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом), операции сравнения модемов по тактовой частоте сигнального процессора, объёму оперативной памяти, количеству портов ввода-вывода, потребляемой мощности. Создать класс **ModemList** (список модемов) в виде массива произвольного числа объектов класса Modem и организовать поиск по марке модема, тактовой частоте сигнального процессора, объёму оперативной памяти, потребляемой мощности.

Вариант 18

Класс **Author** (автор). Поля данных должны включать фамилию автора, имя, отчество, в виде структуры домашний адрес (город, улица, номер дома, номер корпуса и номер квартиры), а также динамический массив строк с названиями литературных произведений. Для строковых данных предусмотреть поля типа char * (С-строка). Реализовать конструктор копирования, конструктор переноса, перегруженные операторы присваивания (с копированием и с переносом). Предусмотреть возможность раздельного изменения составных частей информации об авторе. Создать класс **LitAssociation** (общество литераторов) в виде массива произвольного числа объектов класса Author и организовать поиск автора по какому-либо признаку (например, по количеству произведений или по городу проживания), добавление автора в общество (с помощью перегруженного оператора +), удаление автора из общества (с помощью перегруженного оператора −).